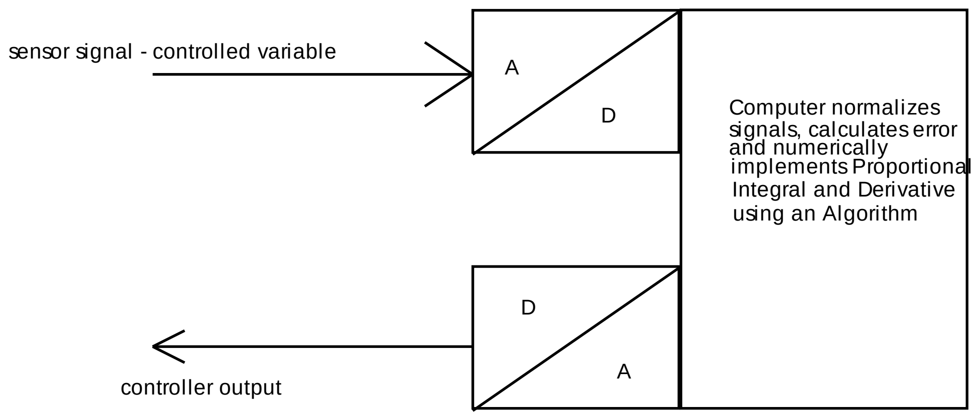
Implementing the PID Using a Computer

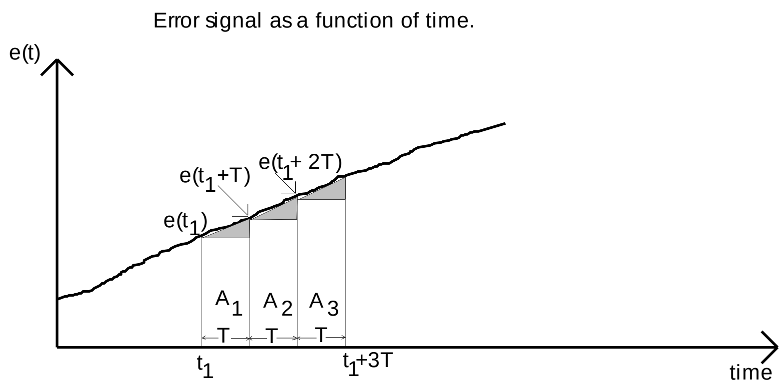


Proportional Control

$C_n = Ke_n$ where n is the sample number and K is the gain of the proportional term. C is the output from the controller.

Integral Control

Shown below is an error signal that varies with time



$A_1 \sim e(t_1)T$
$A_2 \sim e(t_1+T)T$
$A_3 \sim e(t_1+2T)T$

Total Area from t_1 to t_1+3T is $A_{total} = e(t_1)T + e(t_1+T)T + e(t_1+2T)T$
The smaller T is, the more accurate the approximation for total area is.

$$A_{total} = \sum_{i=1}^{3} e(t_i + (i-1)T) \text{ or}$$

or

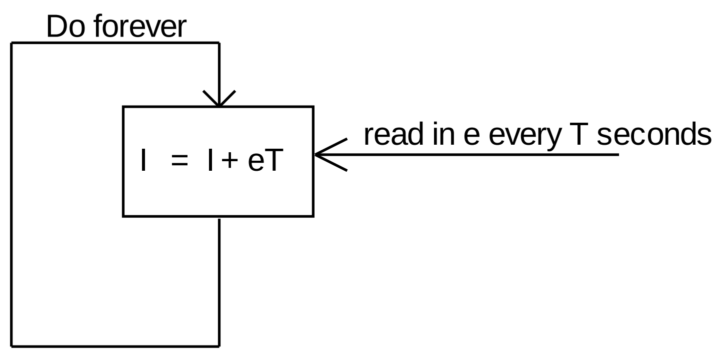
$$A_{total} \cong \int_{t_1}^{t_1+2T} e(t)dt$$

To use a computer to integrate the error, the equation is rewritten in the form of a difference equation. That is:

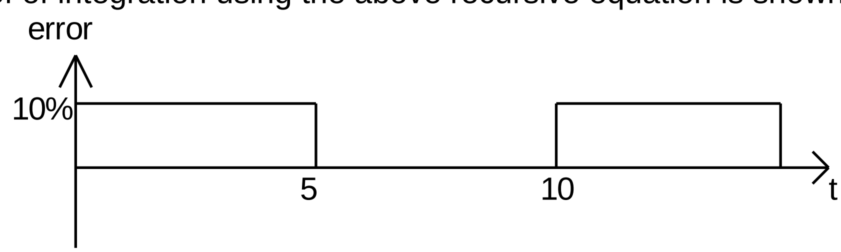
$$I_{n+1} = I_n + e_n T$$

where n is the sample number. I_{n+1} is the new value which is equal to the old value I_n plus the increment $e_n T$.

The difference equation is put into a loop. Thus the equation solution reccurs over and over continuously. The solution is said to be recursive.



An example of of integration using the above recursive equation is shown below:

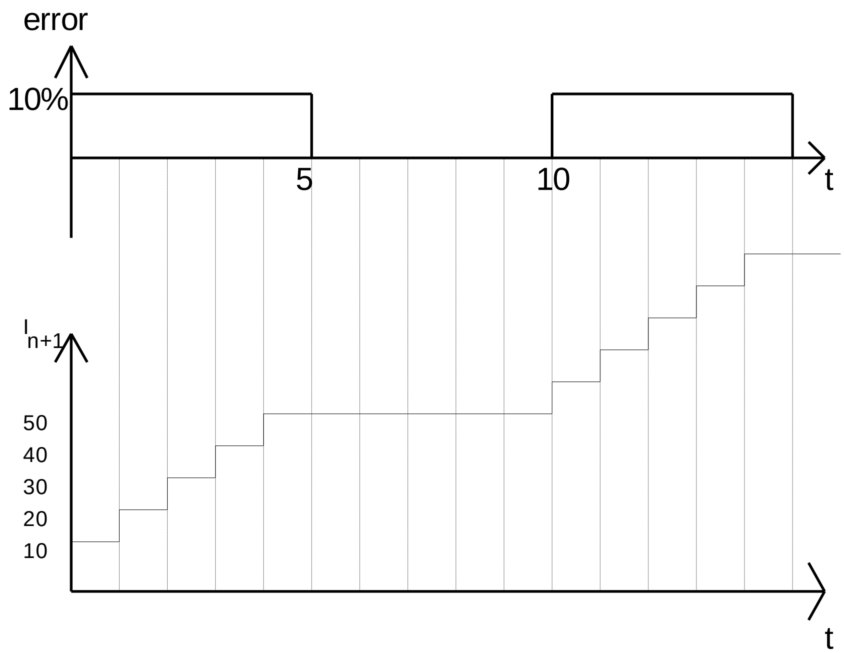


Assume sampling time T is 1 second and that e is sampled at .001 sec, 1.001 sec, 2.001 sec, etc.

Calculate and plot the integral of e recursively using $I_{n+1} = I_n + e_n T$.

Solution

time	.001	1.001	2.001	3.001	4.001	5.001	6.001	7.001	8.001	9.001	10.001
e_n	10%	10%	10%	10%	10%	0%	0%	0%	0%	0%	0%
$I_{n+1}=I_n+e_nT$	10%	20%	30%	40%	50%	50%	50%	50%	50 %	50%	60%



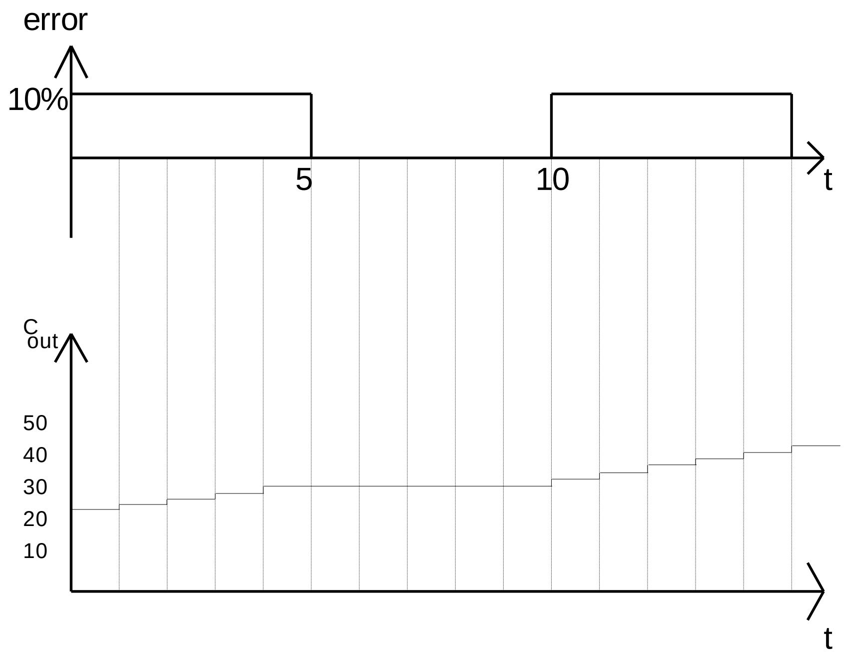
In a PI controller, the integrating term is actually multiplied by the controller gain K and divided by the integral factor T_i. That is:

$$C_{out} = Ke + \frac{K}{T_i}\int edt$$

In the example above, suppose K=2 and T_i=10

Solution

time	.001	1.001	2.001	3.001	4.001	5.001	6.001	7.001	8.001	9.001	10.001
e_n	10%	10%	10%	10%	10%	0%	0%	0%	0%	0%	0%
Ke_n	20%	20%	20%	20%	20%	20%	20%	20%	20%	20%	20%
$I_{n+1}=I_n+(K/T_i)e_n T$	2%	4%	6%	8%	10%	10%	10%	10%	10%	10%	12%
C_{out}	22%	24%	26%	28%	30%	30%	30%	30%	30%	30%	30%



Differentiating Numerically

For $\frac{de}{dt}$ over a small interval of time, de can be approximated by the current value of error minus the old value error where the error is sampled (e_n-e_{n-1}). the dt term can be thought of as the interval between samples, that is the sampling time. For this approximation to be valid the sampling time T must be small compared to the rapidity in the change of the error signal.

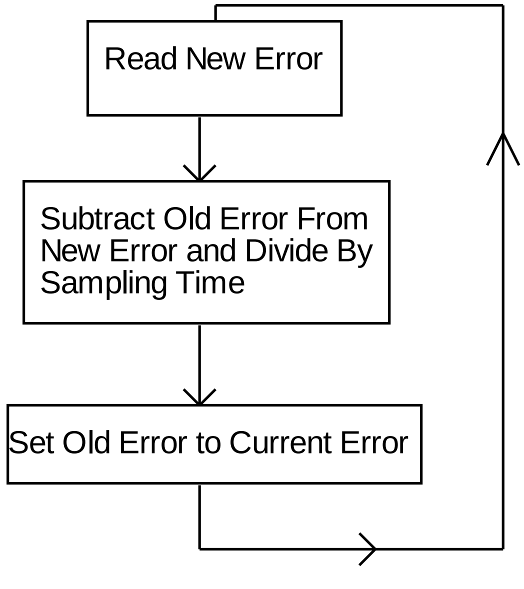
Then $\frac{de}{dt} \approx \frac{e_n - e_{n-1}}{T}$. Remember that the controller output for the PID equation is:

$$C_{out} = K[e + \frac{1}{T_i}\int edt + T_d \frac{de}{dt}]$$

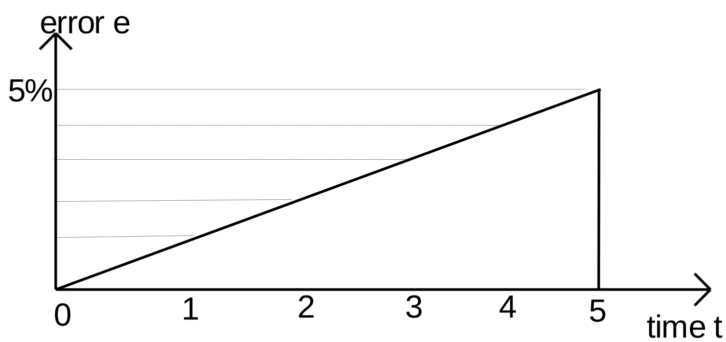
the derivative term of the PID equation can then be expressed as:

$$KT_d(e_n\text{-}e_{n-1})$$

The block diagram for implementing the derivative in a computer real time program would look like:



Example:



Sample time	.001	1.001	2.001	3.001	4.001	5.001	6.001
e_n	0%	1%	2%	3%	4%		
$KT_d \frac{(e_n - e_{n-1})}{T}$							